(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4112420号
(P4112420)
(45)発行日　平成20年7月2日(2008.7.2)　(24)登録日　平成20年4月18日(2008.4.18)

(51)Int.Cl.　F I
G06F 9/50 (2006.01)　G06F 9/46 462A

請求項の数 10 (全 14 頁)

(21)出願番号　特願2003-125028(P2003-125028)
(22)出願日　平成15年4月30日(2003.4.30)
(65)公開番号　特開2003-330734(P2003-330734A)
(43)公開日　平成15年11月21日(2003.11.21)
審査請求日　平成18年3月17日(2006.3.17)
(31)優先権主張番号　10/144,991
(32)優先日　平成14年5月15日(2002.5.15)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT-PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト ハノーバー・ストリート 3000
(74)代理人　100081721
弁理士　岡田　次生
(74)代理人　100105393
弁理士　伏見　直哉
(74)代理人　100111969
弁理士　平野　ゆかり

最終頁に続く

(54)【発明の名称】重みを使用してアプリケーションにシステム資源を割当てる方法およびシステム

(57)【特許請求の範囲】
【請求項１】
コンピュータシステム資源を割当てる方法であって、
システム資源が割当てられる複数のアプリケーションに対して重み比率を計算するステップであって、前記アプリケーションは該アプリケーションに関連付けられた重みを有し、前記重み比率は割当比率と要求比率とを含み、
前記重み比率に基づいて目標比率を設定するステップと、
前記重み比率を使用してアプリケーションの現セットを画定するステップと、
前記目標比率を使用して前記現セットの各アプリケーションに割当てるために十分な未割当資源があるか否かを判断するステップと、
前記目標比率を使用して前記現セットの各アプリケーションに割当てるために十分な未割当資源がある場合、
前記目標比率を新たな目標比率に変更するステップと、
前記新たな目標比率を使用して前記現セットの各アプリケーションに割当てるために十分な未割当資源があるか否か再度判断するステップと、
を含む方法。
【請求項２】
前記目標比率を使用して前記現セットの各アプリケーションに割当てるための未割当資源が十分でない場合、アプリケーションの現セットにおける重みの合計により未割当メモリを除算し、その値を先の目標比率に加算することによって算出された最終比率と現セッ

トにおける各アプリケーションの重みとの積に基づいて資源を割当てるステップをさらに含む請求項１記載の方法。
【請求項３】
前記目標比率に基づく割当後に残っている資源がある場合、最終比率を使用して前記現セットの前記アプリケーションに対し該残っている資源を割当てるステップをさらに含む請求項２記載の方法。
【請求項４】
前記目標比率を新たな目標比率に変更するステップは、該目標比率を次の最も高い重み比率に進めるステップを含む請求項１乃至３のいずれかに記載の方法。
【請求項５】
前記現セットの各アプリケーションに割当てるために十分な未割当資源があるか否かを判断するステップは、各アプリケーションに対する前記目標比率と前記重みとの積に基づいて、前記現セットの各アプリケーションに対しある量の資源を割当てるために十分な未割当資源があるか否かを判断するステップを含む、請求項１乃至４のいずれかに記載の方法。
【請求項６】
コンピュータシステム資源を割当てる方法であって、
複数のアプリケーションに対し、最小使用権の値、最大使用権の値、および重みを含む入力パラメータを受取るステップと、
該パラメータに基づいて前記アプリケーションにシステム資源を割当てるステップと、
を含む方法。
【請求項７】
前記割当てるステップは、
各アプリケーションにその最小使用権を割当てることができるか否かを判断するステップと、
各アプリケーションにその最小使用権を割当てることができる場合、
各アプリケーションにその最小使用権を割当てるステップと、
各アクティブなアプリケーションにその最大使用権を割当てることができるか否かを判断するステップと、
各アクティブなアプリケーションにその最大使用権を割当てることができる場合、
各アクティブなアプリケーションにその最大使用権を割当てるステップと、
各アクティブおよび非アクティブなアプリケーションにその最大使用権を割当てることができるか否かを判断するステップと、
各アクティブおよび非アクティブなアプリケーションにその最大使用権を割当てることができる場合、
各アクティブおよび非アクティブなアプリケーションにその最大使用権を割当てるステップと、
残っている資源を指定されたアプリケーションに割当てるステップと、
を含む請求項６記載の方法。
【請求項８】
各アプリケーションがアクティブであるか非アクティブであるかを判断するステップをさらに含み、該判断するステップと前記割当てるステップとを連続ループにおいて動的に実行し、それによりアプリケーションがアクティブまたは非アクティブとなるにしたがってアプリケーション間の資源の割当を調整する、請求項６または請求項７記載の方法。
【請求項９】
１つまたは複数のアプリケーション間で割当てられるシステム資源を有するコンピュータシステムであって、
メモリと、
アプリケーション間でシステム資源を割当てる方法を実行する命令を実行するプロセッサを有し、該方法は、

複数のアプリケーションに対し、最小使用権の値、最大使用権の値、および重みを含むパラメータを受取るステップと、
該パラメータに基づいて前記アプリケーションにシステム資源を割当てるステップとを含む、
コンピュータシステム。
【請求項１０】
複数のアプリケーション間でコンピュータシステム資源を割当てる方法を実行するコンピュータ実行可能命令が格納された有形のコンピュータ読取可能媒体であって、前記コンピュータ実行可能命令は、
システム資源が割当てられる複数のアプリケーションに対して重み比率を計算するステップと、前記アプリケーションは該アプリケーションに関連付けられた重みを有し、前記重み比率は割当比率と要求比率とを含んでおり、
該重み比率に基づいて目標比率を設定するステップと、
該重み比率を使用してアプリケーションの現セットを画定するステップと、
前記目標比率を使用して前記現セットの各アプリケーションに割当てるために十分な未割当資源があるか否かを判断するステップと、
前記目標比率を使用して前記現セットの各アプリケーションに割当てるために十分な未割当資源がある場合、該目標比率を新たな目標比率に変更し、該新たな目標比率を使用して該現セットの各アプリケーションに割当てるために十分な未割当資源があるか否かを再度判断するステップと、をコンピュータに実行させる、
コンピュータ読取可能媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、一般に、コンピュータシステムに関する。より詳細には、本技術分野は、システムで実行中のアプリケーション間の、メモリ等のシステム資源の割当てに関する。
【０００２】
【従来の技術】
コンピュータシステムの分野では、システム上で実行中のあらゆるプロセスのグループ間でシステム資源を割当てることが望ましい。コンピュータシステムの資源は大抵有限であり、そのシステムのユーザは、それら資源が、最も効率的に使用することができるように特定の方法で割当てられることが確実であるように望む場合がある。ユーザは、グループ間でシステム資源を、直接、または他のシステムプロセスによって割当を確定するシステムを介して間接的に、割当てることができる。また、ユーザは、効率を向上させるためにグループのうちのいくつかまたはすべてに対し最大限の割当分を割当てることができる。
【０００３】
システム資源は、限定されないがメモリ資源、中央処理装置（ＣＰＵ）資源、入出力（Ｉ／Ｏ）帯域幅（ディスクＩ／Ｏ帯域幅またはネットワークＩ／Ｏ帯域幅等）等を含む、コンピュータシステムにおけるいかなる有限の資源をも含む。それらの資源を使用するプロセスのグループは、システム上で実行しているソフトウェアアプリケーション、アプリケーションの一部、またはシステム資源を使用する任意の他のプロセスであってよい。
【０００４】
【発明が解決しようとする課題】
問題は、グループのうちの１つが非アクティブであるかまたは使用可能状態でない場合に発生する。例として、グループによっては、オフラインになった場合、一定時間のみ動作する場合、もしくはそれ以外の方法でオフにされシステム資源を使用していない場合、非アクティブになる場合がある。グループとしてのソフトウェアアプリケーションの場合、一定の時刻にのみ実行するか、もしくは他のあるシステムアクティビティに応答する場合にのみ実行するものもある。アプリケーションの必要にしたがってシステム資源を効率的に割当てる方法およびシステムが必要である。

【０００５】
【課題を解決するための手段】
アプリケーション間におけるメモリ等のコンピュータシステム資源を割当てる方法が開示される。アプリケーションに対し、入力パラメータが受取られる。パラメータは、最小使用権の値と、最大使用権の値と、重みとを含む。最小使用権は、アプリケーションが受取るべき資源の最小量である。最大使用権は、アプリケーションが受取るべき資源の最大量である。重みは、システム資源を分配する目的で、他のアプリケーションに関してアプリケーションの優先度を指定する。コンピュータシステム資源は、それらパラメータに基づいてアプリケーション間で割当てられる。
【０００６】
また、たとえばユーザにより、アプリケーションに対して指定されるパラメータに基づいて、アプリケーション間でメモリ等のコンピュータ資源を割当てる方法も開示する。パラメータは、各アプリケーションに対し、最小使用権と、最大使用権と、重みとを含む。アプリケーションにすでに割当てられた資源の量を重みで除算した値に等しい割当比率と、要求資源の量を重みで除算した値に等しい要求比率とを含む、重み比率が計算される。これらの重み比率を使用して、目標比率が設定され、重み比率に基づいてアプリケーションの現セット（割当作業される組）が画定される。本方法は、目標比率を使用して現セットの各アプリケーションを割当てるために十分な未割当資源があるか否かを判断する。十分な未割当資源がある場合、本方法は、目標比率を新たな目標比率に変更し、新たな目標比率を使用して現セットの各アプリケーションに割当てるために十分な未割当資源があるか否かを再び判断する。
【０００７】
また、メモリおよびプロセッサを有するコンピュータシステムも開示する。プロセッサは、複数のアプリケーション間でシステム資源を割当てる方法を実行する命令を実行する。システムによって実行される方法は、最小使用権と、最大使用権と、重みとを含む、アプリケーションに対するパラメータを受取る。これらのパラメータに基づいて、本方法は、アプリケーション間でシステム資源を割当てる。また、コンピュータシステム資源を割当てる方法を実行するコンピュータ実行可能命令が格納された、有形のコンピュータ読取可能媒体も開示する。
【０００８】
詳細な説明は、以下の図面を参照する。図面において、同様の数字は同様の要素を示している。
【０００９】
【発明の実施の形態】
図１は、システム資源を割当てる方法が実現されるコンピュータシステム１０を示す。コンピュータシステム１０は、オペレーティングシステム５０を使用してアプリケーション２０～２３を実行するプロセッサ３０を含む。プロセッサ３０は、ユーザ入力装置７０から入力信号を受取る。また、プロセッサ３０は、ディスプレイ６０にデータを出力する。コンピュータシステム１０は、メモリ４２、中央処理装置（ＣＰＵ）４４および入出力（Ｉ／Ｏ）帯域幅４６等のシステム資源４０を含む。いかなる時も、各アプリケーション２０～２３は、アクティブであっても非アクティブであってもよい。
【００１０】
本方法およびシステムは、たとえばユーザによりユーザ入力装置７０を介して各アプリケーション２０～２３に対して指定されたパラメータに基づいて、アプリケーション２０～２３にシステム資源４０を割当てる。本明細書で使用する「アプリケーション」という用語は、システム資源にアクセスするプロセスまたはプロセスのグループを言う。アプリケーションは、たとえば、ソフトウェアアプリケーションまたはモジュールを含む。便宜上および単に例として、本明細書では、システム１０を、割当てられる資源４０としてメモリ４２に関して説明する。パラメータは、各アプリケーション（ｇ）（application(g)）に対し、最小使用権min(g)、最大使用権max(g)およびアプリケーションの重みweight(g)

を含む。最小および最大使用権は、アプリケーションに対して許可すべきメモリの最小および最大量をいう。本明細書で使用する「重み」という用語は、異なる重みを有するアプリケーションに対し割当中に異なる量の未割当資源が割当てられるように、システム資源を割当てるために使用する要素を言う。

【００１１】

アプリケーション２０～２３がアクティブおよび非アクティブになると、システム１０は、システム資源４０の分配を調整することにより、それら資源をより効率的に使用する。概して、アプリケーション２０～２３が非アクティブとなると、システム１０は、可能な場合は、そのアプリケーションの資源４０の一部をアクティブなアプリケーションに再び割当てるよう試みる。一実施形態では、システム資源を割当てる方法が、たとえばループにおいて動的に繰返され、それによってシステム１０が、システム資源４０を再割当すべきか否かを連続的に判断する。

【００１２】

図２は、コンピュータシステム１０においてアプリケーション間でメモリを割当てる方法のフローチャートである。すべてのアクティブおよび非アクティブなアプリケーション２０～２３がその最大使用権（max(g)）を受取るために十分なメモリが存在しない限り、メモリは、最小および最大使用権にしたがってアプリケーションの重みに基づいて割当てられる。図２の実施例では、方法は、開始して（ブロック２０９）、各アプリケーション２０～２３にその最小使用権に割当てることができるか否かを判断する（ブロック２１０）。すべてのアプリケーション２０～２３がその最小使用権を受取ることができるとは限らない場合、最初に、メモリはまったく割当てられない（ブロック２９０）。次に、各アプリケーション２０～２３は、変数、request(g)（「要求量」とも言う）をmin(g)に設定し（ブロック２９２）、重みに基づいて利用可能なメモリを割当てる（ブロック３００）ことにより、最小使用権を受取るように試みる。代替実施形態では、システム１０は、ユーザが総システム資源を超過する最小使用権の値を指定するのを可能にせず、そのためステップ２１０における「no」分岐は不要となる。

【００１３】

各アプリケーション２０～２３がその最小使用権（min(g)）を受取ることができる場合（ブロック２１０における「yes」分岐）、本方法は、各アクティブなアプリケーションがその最大使用権（max(g)）を受取るために、および各非アクティブなアプリケーションがその最小使用権（min(g)）を受取るために、利用可能なメモリが十分にあるか否かを判断する（ブロック２２０）。すべてのアクティブなアプリケーションがその最大使用権を受取ることができるとは限らない場合、各アクティブおよび非アクティブアプリケーション２０～２３には、その最小使用権が割当てられる（ブロック２８０）。次に、各アクティブなアプリケーションは、アクティブなアプリケーションに対し変数request(g)をmax(g)に等しい値に設定することによりその最大使用権を要求し、非アクティブなアプリケーションに対しrequest(g)をmin(g)に等しい値に設定する（ブロック２８２）。次に、それらの重みに基づいて、いずれかの残りのメモリがアプリケーションに割当てられる（ブロック３００）。

【００１４】

各アクティブなアプリケーションがその最大使用権を受取ることができる場合（ブロック２２０における「yes」分岐）、本方法は、各アクティブおよび非アクティブなアプリケーション２０～２３がその最大使用権を受取るために、十分なメモリが利用可能であるか否かを判断する（２３０）。すべてのアクティブおよび非アクティブなアプリケーション２０～２３がその最大使用権を受取ることができるとは限らない場合、各アクティブなアプリケーションにはその最大使用権が割当てられ、各非アクティブなアプリケーションにはその最小使用権が割当てられる（ブロック２７０）。すべてのアプリケーションに対し、要求量がmax(g)に等しい値に設定される（ブロック２７２）。残りのメモリは、非アクティブなアプリケーションに対し、それらの重みに基づいて有効に割当てられる（ブロック３００）。

【００１５】
各アクティブおよび非アクティブなアプリケーション２０～２３がその最大使用権を受取ることができる場合（ブロック２３０における「yes」分岐）、アプリケーション２０～２３には、それらの最大使用権が割当てられる（２４０）。方法は、メモリが残っているか否かを判断する（２５０）。メモリが残っていない場合、本方法（３００）は終了する（ブロック３０１）。メモリが残っている場合、図２の方法例は、残っているメモリを指定されたまたはデフォルトのアプリケーションに割当てる（２６０）。他の実現は、異なる方法を使用して、各アプリケーションの最大使用権を満足した後に残っている資源を分配してもよい。

【００１６】
図３は、要求量を使用して、重みに基づいてアプリケーション２０～２３間で残りのメモリを割当てる方法３００の一実現のフローチャートを示す。重み比率が計算され（３１０）メモリを割当てるために使用される。重み比率は、アプリケーションにすでに割当てられたメモリの量をアプリケーションの重み（weight(g)）で除算した値である割当比率と、要求量（request(g)）を重み（weight(g)）で除算した値である要求比率とを含んでよい。

【００１７】
目標比率は、方法３００が資源を割当てるために使用する比率であり、それによって、方法３００は、利用可能な資源を目標比率に比例してアプリケーションに割当てるよう試みる。方法３００は、目標比率に対し低い値で開始し、その目標比率に基づいてメモリを割当てるように試みる。次に、方法３００は、それより高い目標比率に進み、ふたたびメモリを割当てるように試みる。この目標比率を増大させるプロセスは、方法３００がメモリの割当に使用するために大きすぎる目標比率に達するまで続く。図３に示す実施形態では、方法３００は、現目標比率と先の目標比率との両方を追跡する。現目標比率は、それを使用して資源を割当てることができるか否かを判断する試験を目下行っている目標比率である。目標比率が進むと、先の目標比率が格納され方法３００によって使用される。

【００１８】
図３の実施例では、先の目標比率は最初に、最低重み比率、すなわち最低割当比率または要求比率に設定される（ブロック３２０）。現目標比率は、最初に２番目に低い重み率に等しい値に設定される（ブロック３２０）。一実施形態では、２つの重み比率が等しい場合、使用される比率は次に低い値であり、現比率と目標比率とが異なる値を有することになる。たとえば、アプリケーションＡ～Ｃに対して昇順の重み比率が、０、０、０、１０、２０および５０である場合、先の目標比率は、最初に０（最低重み比率として）に等しく設定され、現目標比率は、１０（２番目に低い重み比率として）に等しく設定される。これは、たとえば、各アプリケーション２０～２３にその最小値を割当てるためには資源が不十分であり、アプリケーション２０～２３の各々に対し割当比率が０に設定される場合（図２のブロック２１０における「no」分岐にしたがう）に発生する。図２のフローチャートのこの分岐にしたがって、最初の割当比率はすべて０に等しくなる。これはまた、２つ以上のアプリケーションが共通の重み比率を共有する他の場合にも発生する。

【００１９】
方法３００は、方法３００が適用される際に変化する可能性のある、アプリケーション２０～２３の「現セット」を画定する。アプリケーション２０～２３の現セットは、目標比率より低い割当比率を有し、目標比率以上の要求比率を有するすべてのアプリケーションのセットとして画定される（ブロック３３０）。方法３００は、最低の値で開始して重み比率を上昇させ、その「目標比率」を使用してメモリをアプリケーションの現セットに割当てるよう試みる。目標比率を使用してアプリケーションの現セットに割当てるメモリが十分である場合、目標比率に基づいてメモリが現セットのアプリケーションに割当てられる。目標比率は、次の最も低い重み比率に進み、方法３００は再び新たな目標比率に基づいてメモリを割当てるよう試みる。これは、分配するメモリが不十分となるまで続き、不十分となった時点で、未割当メモリは、現セットの重みの合計に基づいて分配される。

【００２０】
方法３００は、未割当メモリが、アプリケーションの現セットの重みの合計に、現目標比率から先の目標比率を引いたものを乗算したものより、小さいか否かを判断する（ブロック３４０）。未割当メモリがアプリケーションの現セットの重みの合計を超過する場合（ブロック３４０における「no」分岐）、目標比率を使用して現セットの各アプリケーションにメモリを割当てるために十分なメモリがある。現セットにおける各アプリケーションに対し、目標比率にアプリケーションの重み（weight(g)）を乗算した値に等しくメモリが割当てられる（ブロック３５０）。図３に示す実施例では、ブロック３５０において割当てられた量が、先に割当てられた量に置換わる。たとえば、アプリケーションＡに、先に６０単位のメモリが割当てられており、現目標比率がそれに対して合計８０単位のメモリを割当てるというものである場合、先に割当てられた６０の量が８０に置換えられる。他の実施形態では、関数は加法であってよく、それによって本方法は、現割当を置換える量を計算するのではなく、すでに割当てられたメモリに加算するメモリの量を確定する。次に、現目標比率は、すべてのアプリケーションに対し次の最も高い重み比率に進み（ブロック３６０）、方法３００は、ループしてブロック３３０に戻る。
【００２１】
次に、アプリケーションのパラメータに基づいて、先の割当中に最大使用権（max(g)）に達したアプリケーションを除き、次の割当ラウンドにおいてメモリに対して使用権が与えられるアプリケーションを追加して、現セットが再定義される（ブロック３３０）。方法３００は、再び、現セットに対しそれらの重みの合計に基づいて分配するために十分な未割当メモリがあるか否かを判断する（ブロック３４０）。すでに十分な未割当メモリがない場合（ブロック３４０における「yes」分岐）、アプリケーションの現セットにおける重みの合計により未割当メモリを除算し、その値を先の目標比率に加算することによって、最終比率が計算される（３７０）。次に、現セットにおける各アプリケーションに対し、最終比率とその重み（weight(g)）との積に等しいメモリが割当てられる（３８０）。
【００２２】
図４ａ乃至図４ｄは、図２および図３で説明した方法を使用するメモリ割当の計算例を示す。図４ａは、たとえばユーザ入力装置７０から受取ることができる４つのアプリケーションＡ～Ｄに対するアプリケーションパラメータを示す。この実施例では、すべてのアプリケーションがアクティブである。アプリケーションＡ～Ｄは、それぞれ１０、５、５０および１０の最小使用権を有する。また、アプリケーションＡ～Ｄは、それぞれ８０、２０、６０および１００の最大使用権と、それぞれ２、５、２および１の重みとを有している。この実施例では、総利用可能システムメモリは１８５単位である。図２において説明した方法にしたがうと、最小使用権の合計は７５であり、そのため、各アプリケーションＡ～Ｄがその最小使用権（min(g)）を受取るために十分なメモリが存在する（図２のブロック２１０における「yes」分岐）。アプリケーションの最大使用権の合計は（それらはすべてアクティブであるため）２６０であり、それは、１８５の総メモリを超過するため、各アプリケーションはその最大使用権を受取ることができない（図２のブロック２２０における「no」分岐）。
【００２３】
図４ｂは、各アプリケーションＡ～Ｄに対する割当と、request(g)と、割当および要求比率との計算を示す。各アプリケーションＡ～Ｄには、その最小使用権が割当てられており（図２のブロック２８０）、そのためアプリケーションＡ～Ｄの初期割当は、それぞれ１０、５、５０および１０である。未割当メモリは、この時１１０（１８５－７５）である。各アプリケーションＡ～Ｄに対するrequest(g)は、各アプリケーションの最大使用権（max(g)）に等しい値に設定される（すべてのアプリケーションＡ～Ｄがアクティブであるため）（図２のブロック２８２）。アプリケーションＡ～Ｄに対し、requestはそれぞれ８０、２０、６０および１００に等しい値に設定される。最初に、割当比率は、割当（この実施例では各アプリケーションに対するmin(g)）をアプリケーションの重みで除算した値として計算される（図３のブロック３１０）。要求比率は、request（この実施例ではm

ax(g)）をアプリケーションの重みで除算した値として計算される（図３のブロック３１０）。

【００２４】

図４ｃは、図３のフローチャートの下位部分に示す方法によってメモリが割当てられ、方法が、あらゆる目標比率に基づいてメモリを割当てるように試みて、ブロック３３０～３６０として画定されるループを実行する際の、実施例の計算を示す。ループの第１のパスの間、先の目標比率には、割当および要求比率の最低値として１が設定され、現目標比率には、割当および要求比率の２番目に低い値として、４が設定される。この実施例では、現目標比率と先の目標比率との両方が、最初に、同じアプリケーション、すなわちアプリケーションＢから導出される。現セットは、目標比率より低い割当比率と目標比率以下の要求比率とを有するアプリケーションとして画定される（図３のブロック３３０）。この実施例では、第１のパスの間、現セットにはアプリケーションＢのみがある。したがって、現セットの重みの合計は５である。重みの合計と、現目標比率と先の目標比率との差との積は、１５（５×（４－１））である。未割当メモリ（１１０単位）がこの積を超過するため（図３のブロック３４０における「no」分岐）、現目標比率を使用してメモリを現セットに割当てることができる。アプリケーションＢは、２０単位（４×５）のメモリを受取り、アプリケーションＢに先に割当てられた１０と置換わる。図４ｃに示す実施例では、「割当」列の下において、ループのパス中にメモリを受取るアプリケーションは、アスタリスクによって指示されている。アプリケーションＢに対して１５単位を割当てた後、その時の未割当メモリは、９５（１１０単位－１５単位）の値を有する。目標比率は、次の最も高い重み比率の値に進み（図３のブロック３６０）、方法３００は、ブロック３３０に戻ってこの目標比率に基づいて割当を試みる。

【００２５】

ループの第２のパス中、先の目標は４であり、現目標は５である。現セット内でいずれのアプリケーションも適合しないため、第２のパス中にはメモリは割当てられない。目標比率は再び進み（図３のブロック３６０）、方法３００は第２のパスを行う。第３のパス中、現目標は１０であり、先の目標は５である。パス３において画定される現セット内では、アプリケーションＡのみが適合する。アプリケーションＡの重みは２であり、その重みと、現目標比率と先の目標比率との差との積は１０（２×（１０－５））である。次に、アプリケーションＡの割当は２０になり、未割当メモリはこの時総計８５になる。目標比率は次に高い重み比率に進み、方法３００はループの第４のパスを行う。

【００２６】

第４のパス中、現目標比率は２５であり、先の目標比率は１０である。したがって、現セットは、アプリケーションＡおよびＤを含む。現セットの重みの合計は３である。重みと、現目標と先の目標との差と、の積は４５である。この量は、未割当メモリより小さく、そのため現目標比率を使用してメモリがアプリケーションＡおよびＤに割当てられる。目標比率に重みを乗算することにより、アプリケーションＡの割当は５０になり、アプリケーションＤの割当は２５になる。未割当メモリは、この時４０である。現目標比率が進み、方法３００はループの第５のパスを行う。

【００２７】

第５のパス中、現目標比率は３０であり、先の目標比率は２５である。この時、現セットはアプリケーションＡ、ＣおよびＤを含む。現セットの重みの合計は、５（２＋２＋１）であり、重みの合計と、現目標比率と先の目標比率との差との積は２５である。この値は、未割当メモリより小さく、そのため現比率を使用してメモリＡ、ＣおよびＤにメモリが割当てられる。ループの第５のパスの後、アプリケーションＡ～Ｄに、それぞれ６０、２０、６０および３０単位のメモリが割当てられる。未割当メモリは、総計１５となる。目標比率は進み、方法はループの第６のパスに入る。

【００２８】

第６のパス中、現目標比率は４０であり、先の目標比率は３０である。現セットは、AおよびＤである。アプリケーションＣは、先の割当ラウンド中にその最大メモリ使用権の６

０に達しており、したがって、第６のパス中の現セットから落とされる。重みの合計は３であり、この合計と、現目標比率と先の目標比率との差との積は３０であり、そのためパスは３０単位の未割当メモリを必要とする。未割当メモリが３０より小さい場合（図３のブロック３４０における「yes」分岐）、現目標比率を使用することができず、最終比率が計算されて（図３のブロック３７０）未割当メモリの残りの１５単位が割当てられる。

【００２９】

図４ｄは、最終比率の計算とその割当とを示す。最終比率は、先の目標比率に、未割当メモリを現セットの重みの合計で除算した値を足したものである（３０＋（１５／３）＝３５）。最終比率を使用して、未割当資源が現セットに、その中のアプリケーションの重みにしたがって割当てられる。この実施例における最終割当（使用権すなわちentitlement(g)とも言う）は、それぞれのグループに対して７０、２０、６０および３５である。

【００３０】

図５ａないし図５ｄは、図４ａないし図４ｄの実施例に類似する、図２および図３において説明する方法を使用するメモリ割当のさらなる計算例を示すが、この場合、アプリケーションのうちの１つ、アプリケーションＡが非アクティブである。アプリケーションＡが非アクティブであるため、そうでなければアプリケーションＡに割当てられるメモリ資源の一部が、アクティブなアプリケーションに再割当される。図５ａは、アクティブかまたは非アクティブないずれかとしてアプリケーションのステータスのインジケータを含む、アプリケーションパラメータの図表を示す。図５ａないし図５ｄの実施例は、アプリケーションＡが非アクティブであるということを除いて、図４ａないし図４ｄの実施例に従うため、アプリケーションの最小使用権（min(g)）と、最大使用権（max(g)）と、重み（weight(g)）との値は、図４ａに示すものと同じである。

【００３１】

図５ｂは、ここでまた図２および図３に関して説明した方法を使用する、各アプリケーションに対する割当と、request(g)と、割当および要求比率との計算を示す。図４ｂの実施例に関し、各アプリケーションにその最小使用権が割当てられ（図２のブロック２８０）、そのためアプリケーションＡ～Ｄの最初の割当は、それぞれ１０、５、５０および１０である。未割当メモリは、この時１１０（１８５－７５）である。各アクティブなアプリケーションに対する要求は、最大使用権に等しい値に設定される（図２のブロック２８２）。この実施例ではアプリケーションＡが非アクティブであるため、その要求量は、その最小使用権の１０に設定され、それはすでに割当てられている。アプリケーションＡ～Ｄに対し、要求がそれぞれ１０、２０、６０および１００に等しい値に設定される。最初に、割当比率が、最小使用権をアプリケーションの重みで除算した値として計算される（ブロック３１０）。要求比率は、要求をアプリケーションの重みで除算した値として計算される（ブロック３１０）。

【００３２】

図５ｃは、メモリが図３のフローチャートの下位部分に示す方法によって割当てられ、方法があらゆる目標比率に基づいてメモリを割当てるように試みてループを実行する（図３のブロック３３０～３６０）際の、実施例の計算を示す。ループの第１のパスは、図４ｃに示すものと実質的に同じである。図３のブロック３３０～３６０として画定されるループの第１のパス中、先の目標比率は、割当および要求比率の最低値として１に設定され、現目標比率は、割当および要求比率の２番目に低い値として４に設定される。第１のパス中、現セットにはアプリケーションＢのみがある。現セットの重みの合計は５である。重みの合計と、現目標比率と先の目標比率との差との積は、１５（５×（４－１））である。未割当メモリ（１１０単位）がこの積（１５単位）を超過するため（図３のブロック３４０の「no」分岐）、現目標比率を使用して、メモリを現セットに割当てることができる。アプリケーションＢは、２０単位（４×５）のメモリを受取って、アプリケーションＢに先に割当てられた１０単位に置換わる。この時、未割当メモリは、９５（１１０単位－１５単位）の値を有する。目標比率は、次に高い重み比率の値に進み（図３のブロック３６０）、方法３００は、ブロック３３０に戻って、この目標比率に基づいて割当てるよう

に試みる。

【００３３】

第２のパス中、先の目標は４であり、現目標は５である。その要求比率が、アプリケーションＡが非アクティブであるために減少したため、この時アプリケーションＡは、この実施例では第２のパス中に現セットに適合する。現セットの重みの合計は２であり、重みと目標比率との積は１０である。しかしながら、アプリケーションＡは、すでに１０単位のメモリを有しているため、このパス中にそれ以上メモリを受取らず、第２のパス後に未割当資源は９５のままである。第３のパス中、目標比率は、次の重み比率１０まで増大する。第３のパス中の現セット内にいずれのアプリケーションも適合しないため、目標比率は、第４のパスのために２５まで増大する。

【００３４】

第４のパス中、現目標比率は２５であり、先の目標比率は１０である。したがって、アプリケーションの現セットは、アプリケーションＤのみを含む。現セットの重みの合計は１である。現セットの重みの合計と、現目標と先の目標との差と、の積は１５である。この量は、未割当メモリより小さく、そのため現目標比率を使用してメモリがアプリケーションＤに割当てられる。目標比率に重みを乗算することにより、アプリケーションＤの割当は２５になる。未割当メモリは、この時８０である（図４ｄの実施例における４０ではなく）。現目標比率が進み、方法３００はループの第５のパスを行う。

【００３５】

第５のパス中、現目標比率は３０であり、先の目標比率は２５である。この時、現セットは、アプリケーションＣおよびＤを含む。現セットの重みの合計は、３（２＋１）であり、重みの合計と、現目標比率と先の目標比率との差との積は１５である。この値は、未割当メモリより小さく、そのため、現比率を使用してメモリがＣおよびＤに割当てられる。ループの第５のパス後、アプリケーションＡ～Ｄに、それぞれ１０、２０、６０および３０単位のメモリが割当てられる。未割当メモリは、総計６５となる。目標比率が進み、方法３００はループの第６のパスに入る。

【００３６】

第６のパス中、現目標は１００であり、先の目標は３０である。アプリケーションＣは、先のラウンド中にその最大メモリ使用権６０に達したため、現セットはＤのみである。重みの合計は１であり、この合計と、現目標比率と先の目標比率との差との積は７０であり、そのためこのパスは７０単位の未割当メモリを必要とする。未割当メモリが７０より少ないため（図３のブロック３４０における「yes」分岐）、現目標比率を使用することができず、最終比率を計算することによって（図３のブロック３７０）、残りの６５単位の未割当メモリが割当てられる。

【００３７】

図５ｄは、最終比率の計算とその割当とを示す。最終比率は、先の目標比率に、未割当メモリを現セットの重みの合計で除算した値を足したものである（３０＋（６５／１）＝９５）。最終比率を使用して、アプリケーションの重みにしたがって未割当資源が現セットに割当てられる。この実施例における最終割当（使用権またはentitlement(g)とも言う）は、それぞれのグループに対して１０、２０、６０および９５である。

【００３８】

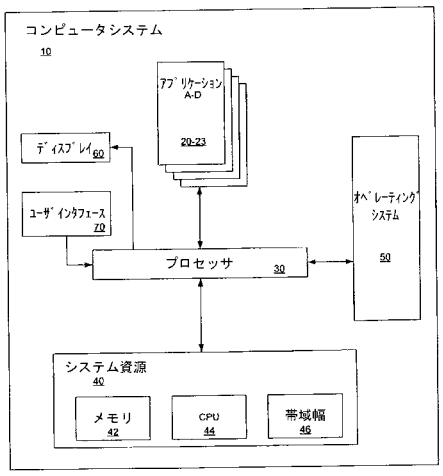
本発明を、その特定の実施形態に関して説明したが、変形が可能である。たとえば、資源を割当てる方法を、メモリを割当てることに関して説明したが、当業者は、それがあらゆるタイプのシステム資源に割当てられてよい、ということを認めるであろう。本発明を、その本質的精神または特性から逸脱することなく、特定の形態で実現してよい。本明細書で説明した実施形態が、すべての点で限定的ではなく例示的であるようにみなされ、本発明の範囲を確定するために併記の特許請求の範囲とそれらの等価物とが参照されることが望ましい。

【図面の簡単な説明】

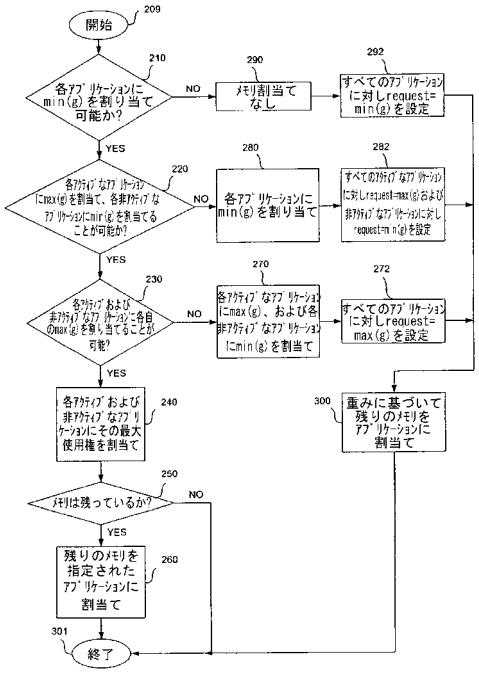
【図１】システム資源を割当てる方法が実現されるコンピュータシステム。

【図２】メモリを割当てる方法のフローチャート。
【図３】図２のステップ３００に示すように、重みに基づいてメモリを割当てる方法のフローチャート。
【図４ａ】メモリ割当の計算例。
【図４ｂ】メモリ割当の計算例。
【図４ｃ】メモリ割当の計算例。
【図４ｄ】メモリ割当の計算例。
【図５ａ】メモリ割当のさらなる計算例。
【図５ｂ】メモリ割当のさらなる計算例。
【図５ｃ】メモリ割当のさらなる計算例。
【図５ｄ】メモリ割当のさらなる計算例。

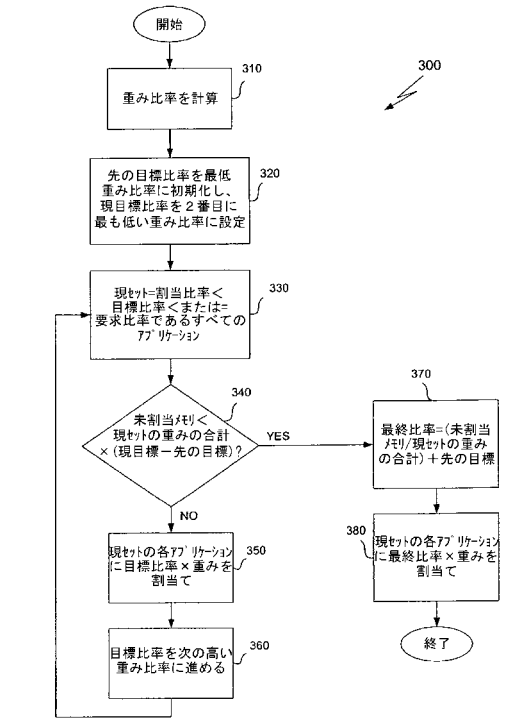
【図１】



【図２】



【図３】



【図４ａ】

ｱﾌﾟﾘｹｰｼｮﾝﾊﾟﾗﾒｰﾀ　--すべてのｱﾌﾟﾘｹｰｼｮﾝがｱｸﾃｨﾌﾞ

ｱﾌﾟﾘｹｰｼｮﾝ	Min(g)	Max(g)	Weight(g)
A	10	80	2
B	5	20	5
C	50	60	2
D	10	100	1

【図４ｂ】

総資源 = 185

ｱﾌﾟﾘｹｰｼｮﾝ	割当	Request(g)	割当比率	要求比率
A	10	80	5	40
B	5	20	1	4
C	50	60	25	30
D	10	100	10	100

【図４ｃ】

パス	先の目標比率	現目標比率	現セット	現セットの重みの合計	重みと現目標比率との差		割当	未割当資源
1	1	4	B	5	15	A	10	95
						B	20*	
						C	50	
						D	10	
2	4	5	0	0	0	A	10	95
						B	20	
						C	50	
						D	10	
3	5	10	A	2	10	A	20*	85
						B	20	
						C	50	
						D	10	
4	10	25	AD	3	45	A	50*	40
						B	20	
						C	50	
						D	25*	
5	25	30	ACD	5	25	A	60*	15
						B	20	
						C	60*	
						D	30*	
6	30	40	AD	3	30			

【図４ｄ】

最終比率 = 30 + 15/3 = 35

	最終割当
A	70*
B	20
C	60
D	35*

【図５ａ】

ｱﾌﾟﾘｹｰｼｮﾝﾊﾟﾗﾒｰﾀ

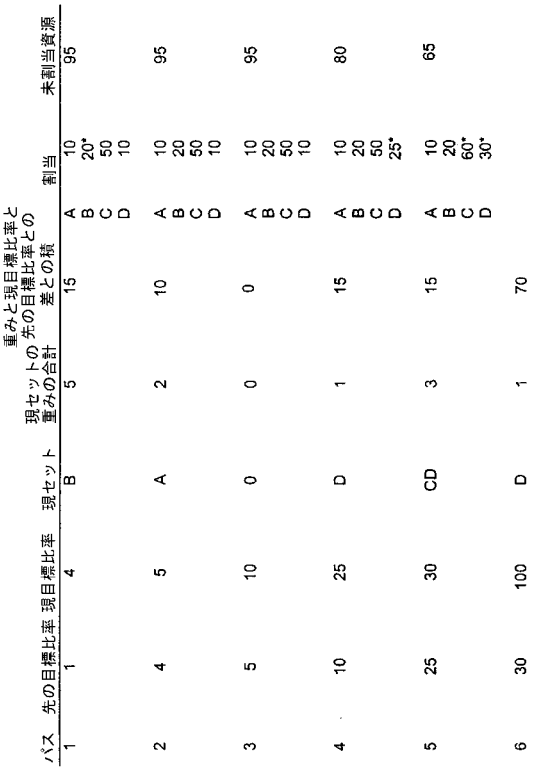
ｱﾌﾟﾘｹｰｼｮﾝ	ｽﾃｰﾀｽ	Min(g)	Max(g)	Weight(g)
A	Inactive	10	80	2
B	Active	5	20	5
C	Active	50	60	2
D	Active	10	100	1

【図５ｂ】

総資源 = 185

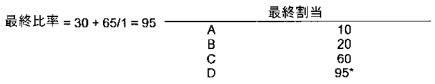
ｱﾌﾟﾘｹｰｼｮﾝ	割当	Request(g)	割当比率	要求比率
A	10	10	5	5
B	5	20	1	4
C	50	60	25	30
D	10	100	10	100

【図５ｃ】



パス	先の目標比率	現目標比率	現セット	現セットの重みの合計	重みと現目標比率と先の目標比率との差との積	割当	未割当資源
1	1	4	B	5	15	A 10	95
						B 20*	
						C 50	
						D 10	
2	4	5	A	2	10	A 10	95
						B 20	
						C 50	
						D 10	
3	5	10	0	0	0	A 10	95
						B 20	
						C 50	
						D 10	
4	10	25	D	1	15	A 10	80
						B 20	
						C 50	
						D 25*	
5	25	30	CD	3	15	A 10	65
						B 20	
						C 60*	
						D 30*	
6	30	100	D	1	70		

【図５ｄ】



最終比率 = 30 + 65/1 = 95

	最終割当
A	10
B	20
C	60
D	95*

フロントページの続き

(72)発明者　クリフォード・エイ・マッカーシー
アメリカ合衆国７５０８０テキサス州リチャードソン、フォレスト・グローヴ・ドライヴ　４０３
(72)発明者　キャンミン・ジン
アメリカ合衆国７５０２５テキサス州プラノ、レイヴンハースト・ドライヴ　２２１７

審査官　鈴木　修治

(56)参考文献　特開２００１－１９５２６８（ＪＰ，Ａ）
特開２００１－３３１３３２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　9/46-9/54
G06F 15/16-15/177